

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2000-32469

(P2000-32469A)

(43) 公開日 平成12年1月28日 (2000.1.28)

(51) Int.Cl.

H 04 N 7/32

識別記号

F I

H 04 N 7/137

テマコード(参考)

Z 5 C 0 5 9

審査請求 有 請求項の数 6 O.L (全 16 頁)

(21) 出願番号 特願平10-197500

(22) 出願日 平成10年7月13日 (1998.7.13)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 上田 裕明

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100065385

弁理士 山下 穂平

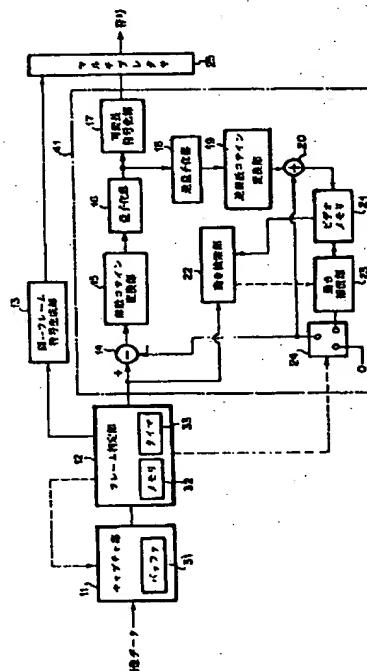
F ターム(参考) 50059 KK00 MA00 MA05 MA14 MA23  
MC11 ME01 NN01 NN21 NN28  
PP05 PP06 PP07 SS11 TA00  
TA18 TA25 TC00 TC01 TC18  
TD01 TD07 TD11 UA02 UA32  
UA33

(54) 【発明の名称】 動画像符号化装置

(57) 【要約】

【課題】 装置の性能を無駄にすることなく高能率にリアルタイムで動画像を圧縮することができる動画像符号化装置を提供する。

【解決手段】 動画像の一部のフレームをフレーム内符号化し、該フレーム内符号化されたフレームに挟まれるフレームの一部を前方フレーム間符号化し、フレーム内符号化されたフレーム (I ピクチャ) 又は前方フレーム間符号化されたフレーム (P ピクチャ) と I ピクチャ又は P ピクチャとの間に挟まれるフレーム (B ピクチャ) を双方向フレーム間符号化する動画像符号化装置において、I ピクチャと P ピクチャのみを符号化したときに装置が単位時間に処理できるピクチャ数である実質フレームレートと出力フレームレートを基に B ピクチャの数を決定する決定手段と、B ピクチャを差分が無いフレームであると扱って符号化する符号化手段とを備える。



1

## 【特許請求の範囲】

【請求項1】 動画像の一部のフレームをフレーム内符号化し、該フレーム内符号化されたフレームに挟まれるフレームの一部を前方フレーム間符号化し、前記フレーム内符号化されたフレーム（Iピクチャ）又は前記前方フレーム間符号化されたフレーム（Pピクチャ）と前記フレーム内符号化されたフレーム（Iピクチャ）又は前記前方フレーム間符号化されたフレーム（Pピクチャ）との間に挟まれるフレーム（Bピクチャ）を双方方向フレーム間符号化する動画像符号化装置において、IピクチャとPピクチャのみを符号化したときに装置が単位時間に処理できるピクチャ数である実質フレームレートと出力フレームレートを基にBピクチャの数を決定する決定手段と、Bピクチャを差分が無いフレームであると扱って符号化する符号化手段とを備えることを特徴とする動画像符号化装置。

【請求項2】 請求項1に記載の動画像符号化装置において、前記符号化手段は前記Bピクチャの符号列として前記動画像のフレームの大きさにより決まる符号列を生成することを特徴とする動画像符号化装置。

【請求項3】 請求項1又は2に記載の動画像符号化装置において、前記決定手段は、装置が最大限に稼働するように前記Bピクチャの数を決定することを特徴とする動画像符号化装置。

【請求項4】 請求項1乃至3のいずれか1項に記載の動画像符号化装置において、前記決定手段は、前記Iピクチャ又はPピクチャと前記Iピクチャ又はPピクチャの間のBピクチャの数が最大限に揃うように前記Bピクチャの数を決定することを特徴とする動画像符号化装置。

【請求項5】 請求項1乃至4のいずれか1項に記載の動画像符号化装置において、前記実質フレームレートと出力フレームレートとの最小公倍数が最小となるような出力フレームレートを複数の所定の出力フレームレートから選択する手段を備えることを特徴とする動画像符号化装置。

【請求項6】 請求項1乃至5のいずれか1項に記載の動画像符号化装置において、前記実質フレームレートを最後に実質フレームレートを更新してから圧縮したフレーム数と、現在の出力フレームレートと、最後に実質フレームを更新してから挿入したBピクチャの数と、最後にフレームレートを更新してからのキャプチャ待ち時間の合計を基に算出して更新する手段を更に備えることを特徴とする動画像符号化装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は動画像を圧縮して符号化する動画像符号化装置に関し、特に、動画像の一部のフレームをフレーム内符号化し、フレーム内符号化されたフレームに挟まれるフレームの一部を前方フレーム

2

間符号化し、フレーム内符号化されたフレーム（Iピクチャ）又は前方フレーム間符号化されたフレーム（Pピクチャ）とフレーム内符号化されたフレーム（Iピクチャ）又は前方フレーム間符号化されたフレーム（Pピクチャ）との間に挟まれるフレーム（Bピクチャ）を双方方向フレーム間符号化する動画像符号化装置に関する。

## 【0002】

【従来の技術】 通常、画像をデジタル化して、CD-R OMやハードディスクなどの記録媒体に記録する場合、10 そのデータ量は膨大なものとなるため、通常は圧縮符号化して記録される。この圧縮符号化方法には、各種の圧縮符号化方式があり、特に、画像の空間周波数が低周波に集中する性質を利用して圧縮を行うDCT（Discrete Cosine Transform）をベースとした符号化方式が、比較的多く使用されている。DCTはJPEG（Joint Photographic Coding Experts Group）や、MPEG（Moving Picture Coding Experts Group）1またはMPEG2などの国際標準の符号化方式の中で使用されている。

## 【0003】

20 【発明が解決しようとする課題】 MPEG1に代表されるような、蓄積系の動画像圧縮方法では、フレームレート（単位時間あたりのフレーム数）が決められている。そのため、圧縮性能の低い動画像符号化装置でもリアルタイムに圧縮するためにフレームレートを落として圧縮することができない。

【0004】 このMPEG1の規格を守りながら実質的なフレームレートを減らす方法として特開平10-42295号公報に記載がある。この方法はフレーム間圧縮を行うフレームが差分がないフレームであることを示す符号を生成することで実際の圧縮処理を省いて実質的なフレームレートを減らしている。

【0005】 しかし、差分がないフレームであることを示す符号を生成する周期が動画像符号化装置の実質的なフレームレートに合うとは限らないので、動画像符号化装置の性能を無駄にすることなく高能率に動画像を圧縮できないという問題点がある。

【0006】 また、特開平10-70727号公報では圧縮ビデオ情報の伝送に遅延が生じた場合にBブロックを削除することにより、遅延を解消している。

40 【0007】 しかし、伝送の遅延と圧縮性能は必ずしも比例しているわけでは無いので、動画像符号化装置の性能を無駄にすることなく高能率に動画像を圧縮することができないという問題点がある。

【0008】 また、特開平9-130787号公報では、実際の符号量に応じてスキップするフレームを決めてフレームレートを調整している。

【0009】 しかし、符号量と圧縮性能は必ずしも比例しているわけでは無いので、動画像符号化装置の性能を無駄にすることなく高能率に動画像を圧縮できないという問題点がある。

50

【0010】本発明は、装置の性能を無駄にすることなく高能率にリアルタイムで動画像を圧縮することができる動画像符号化装置を提供することを目的とする。

【0011】

【課題を解決するための手段】本発明による動画像符号化装置は、動画像の一部のフレームをフレーム内符号化し、該フレーム内符号化されたフレームに挟まれるフレームの一部を前方フレーム間符号化し、前記フレーム内符号化されたフレーム(Iピクチャ)又は前記前方フレーム間符号化されたフレーム(Pピクチャ)と前記フレーム内符号化されたフレーム(Iピクチャ)又は前記前方フレーム間符号化されたフレーム(Pピクチャ)との間に挟まれるフレーム(Bピクチャ)を双方向フレーム間符号化する動画像符号化装置において、IピクチャとPピクチャのみを符号化したときに装置が単位時間に処理できるピクチャ数である実質フレームレートと出力フレームレートを基にBピクチャの数を決定する決定手段と、Bピクチャを差分が無いフレームであると扱って符号化する符号化手段とを備えることを特徴とする。

【0012】また、本発明による動画像符号化装置は、上記の動画像符号化装置において、前記符号化手段は前記Bピクチャの符号列として前記動画像のフレームの大きさにより決まる符号列を生成することを特徴とする。

【0013】更に、本発明による動画像符号化装置は、上記の動画像符号化装置において、前記決定手段は、装置が最大限に稼働するように前記Bピクチャの数を決定することを特徴とする。

【0014】更に、本発明による動画像符号化装置は、上記の動画像符号化装置において、前記決定手段は、前記Iピクチャ又はPピクチャと前記Iピクチャ又はPピクチャの間のBピクチャの数が最大限に揃うように前記Bピクチャの数を決定することを特徴とする。

【0015】更に、本発明による動画像符号化装置は、上記の動画像符号化装置において、前記実質フレームレートと出力フレームレートとの最小公倍数が最小となるような出力フレームレートを複数の所定の出力フレームレートから選択する手段を備えることを特徴とする。

【0016】更に、本発明による動画像符号化装置は、上記の動画像符号化装置において、前記実質フレームレートを最後に実質フレームレートを更新してから圧縮したフレーム数と、現在の出力フレームレートと、最後に実質フレームを更新してから挿入したBピクチャの数と、最後にフレームレートを更新してからのキャプチャ待ち時間の合計を基に算出して更新する手段を更に備えることを特徴とする。

【0017】

【発明の実施の形態】本発明は、動画像をリアルタイムで圧縮符号化する動画像符号化装置において、圧縮性能が低くてもリアルタイムに圧縮符号化ができるように一部のフレームが差分がないフレームであることを示す符

号を挿入して装置の可動サイクルである実質的なフレームレートを下げるものである。

【0018】図1は、本発明の実施形態による動画像符号化装置の構成を示すブロック図である。

【0019】図1において、キャプチャ11が1フレーム毎に画像をキャプチャして、フレーム判定部12が予め圧縮性能と画像サイズから予測した実質フレームレートに最適なGOP構成を求め、それに基づいて現在のフレームのピクチャ種別を判定する。符号化部41はIピクチャとPピクチャの場合は圧縮を行い、Bピクチャの場合は符号化部41は実際の圧縮を行わず、同一フレーム符号生成部13が差分がないフレームであることを示す符号を挿入する。マルチプレクサが符号化部41が出力する符号と同一フレーム符号生成部13が生成する符号を連結する。また、ある一定数のフレームを処理する毎にキャプチャ部11からのキャプチャ待ちの合計時間と挿入したBピクチャのフレーム数等を調べて、その値から実質フレームレートを再計算して、そのフレームレートを基にGOP構成を最適となるように更新する。

【0020】MPEG1などのように単位時間あたりのフレーム数が規定されている圧縮方式を図12、図13を参照して説明する。

【0021】図12はMPEG1ビデオに準拠した符号フォーマットの階層図を示した図である。MPEG1ビデオの符号は図12に示すようにいくつかの階層より構成される階層構造となっている。ビデオデータは1個以上のビデオ・シーケンス(VSC)から構成されていて、ビデオシーケンスエンドコード(VSE)で終了する。ビデオ・シーケンスはビデオシーケンスヘッダ(VSH)と、1個以上のGOP(Group Of Picture)から構成される。GOPは1個以上のピクチャから構成され、1つのピクチャが1フレームを示している。ピクチャにはフレーム内符号のみから構成されるIピクチャと、前方向のみのフレーム間符号から構成されるPピクチャと、前後の双方向のフレーム間符号から構成されるBピクチャの3種類がある。ピクチャは任意の領域に分割された複数のスライスから構成される。スライスは左から右へ、または上から下への順序で並んだ複数のマクロブロックから構成される。マクロブロックには大別してフレーム内符号であるイントラブロックと前方向や双方向のフレーム間符号であるインターブロックの2種類がある。Iピクチャはイントラブロックのみで構成されるが、PピクチャやBピクチャはインターブロックのみでなくイントラブロックも含む場合がある。マクロブロックは $16 \times 16$ 画素のブロックを更に4つの $8 \times 8$ 画素のブロック(Y1, Y2, Y3, Y4)に分割した輝度成分と輝度成分の $16 \times 16$ 画素に一致する領域の $8 \times 8$ 画素のブロックの色差成分(Cb, Cr)から構成される。従って、1つのマクロブロックは、4つの輝度成分のブロックと2つの色差成分のブロックより構成さ

れ、合計で6個のブロックから構成される。 $8 \times 8$ 画素のブロックが符号化の最小単位となる。

【0022】図13はMPEG1のGOP構成を示した図である。フレームのタイプにはI、P、Bがあり、Iピクチャは参照フレームを用いないで圧縮されるフレームであり、Pピクチャは前のIもしくはPピクチャのいずれかを参照して圧縮されるフレームである。Bピクチャは前後のIもしくはPピクチャを参照して圧縮されるフレームであり、Bピクチャ自身は参照フレームとして用いられることはない。MPEG1では単位時間あたりのフレーム数が規定されているため、圧縮処理を行わないフレームというものはあり得ないが、参照フレームとして用いられないBピクチャのデータを参照フレームと全く同じ画像として扱うことで圧縮処理の負荷が非常に小さくなるため、圧縮処理を行わない場合と同様の効果を得ることができる。

【0023】図1は本発明の実施形態による動画像符号化装置の構成図である。図1を参照すると、本実施形態の動画像符号化装置は、原画像をキャプチャするキャプチャ部11と、キャプチャ部11でキャプチャしたフレームを一時的に記憶しキャプチャ部11内に設けられるバッファ31と、処理するフレームのピクチャ種別を圧縮性能に応じたGOP構成を基に判定するフレーム判定部12と、フレーム判定部12で使用する制御変数などを格納しフレーム判定部12内に設けられるメモリ32と、キャプチャ待ちの時間を計測しフレーム判定部12内に設けられるタイマ33と、マクロブロック単位で画像の動きを検索する動き検索部22と、 $8 \times 8$ 画素のブロックの画像を空間周波数成分に変換する離散コサイン変換部15と、空間周波数成分を量子化する量子化器16と、量子化された空間周波数成分を逆量子化する逆量子化18と、逆量子化された空間周波数成分を逆離散コサイン変換する逆離散コサイン変換部19と、マクロブロック単位で参照フレームの画像の動きを補償する動き補償部23と、量子化された空間周波数成分を可変長符号化する可変長符号化部17と、可変長符号化部17が生成する符号と同一フレーム符号生成部13が生成する符号とを連結するマルチプレクサ25とを備える。

【0024】上記の動画像符号化装置の各部はそれぞれ以下のように動作する。

【0025】キャプチャ部11はフレーム判定部12からの指示により指定されたフレームの画像をバッファ31から取り出す。フレーム判定部12は、予め圧縮性能と画像サイズから予測した実質フレームレートに最適なGOP構成から1フレーム単位に入力される現在のフレームのピクチャ種別を判定する。符号化部41は、IピクチャとPピクチャをピクチャ種別に応じて圧縮する。

【0026】現在のフレームがIピクチャである場合は、フレーム判定部12から出力された画像データは、離散コサイン変換部15で $8 \times 8$ 画素単位で空間周波数

成分に変換され、空間周波数成分は、量子化器16で量子化され、量子化された空間周波数成分は、可変長符号化部17で可変長符号化される。可変長符号化部17から出力される可変長符号はマルチプレクサ25に出力され、フレーム判定部12が出力するピクチャタイプなどと連結される。また、量子化された空間周波数成分は逆量子化器18で逆量子化され、逆量子化された $8 \times 8$ ブロックの空間周波数成分は逆離散コサイン変換部19で空間位置成分に戻され、ビデオメモリ21に書き込まれる。

【0027】現在のフレームがPピクチャである場合は、フレーム判定部12から出力された現在のフレームのマクロブロックに最も相関が高いビデオメモリ21に記憶されている参照フレームのマクロブロックが動き検出部22で求められ、その最も相関が高いマクロブロックが動き補償部23から出力され、スイッチ24を通り、スイッチから出力される。フレーム判定部12から出力される現在のフレームの $8 \times 8$ 画素のブロックの各画素データから、スイッチ24を通った参照フレームの最も相関が高いマクロブロック内の $8 \times 8$ 画素のブロックの各画素データが減算器14で減算され、減算器14の出力が離散コサイン変換部15で空間周波数成分に変換される。空間周波数成分は、量子化器16で量子化され、量子化された空間周波数成分は、可変長符号化部17で可変長符号化される。可変長符号化部23から出力される可変長符号はマルチプレクサ25に出力され、同一フレーム符号生成部13が出力するピクチャタイプなどの符号列と連結される。また、量子化された空間周波数成分は逆量子化器18で逆量子化され、逆量子化された $8 \times 8$ ブロックの空間周波数成分は逆離散コサイン変換部19で各画素データに戻され、各画素データに加算器20でスイッチ24の出力が加算され新たな参照フレームのデータとなり、加算器20の出力がビデオメモリ21に書き込まれる。

【0028】現在のフレームがBピクチャである場合は、実際の圧縮を行わずに差分がないフレームであることを示す符号を挿入する。すなわち、フレーム判定部12が現在のフレームがBピクチャであることを判定すると、同一フレーム符号生成部13は、差分がないフレームであることを示す符号列をマルチプレクサ25に出力し、マルチプレクサ25はこの符号列を選択する。また、ある一定数のフレームを処理する毎にフレーム判定部12のタイマ33で計測したキャプチャ時の待ち時間の合計を求めて、その値から実質フレームレートを再計算して、実質フレームレートと出力フレームレートとが整合するようにGOP構成を最適化して更新する。

【0029】次に、図2のフローチャートを参照して本実施形態の全体の動作について詳細に説明する。

【0030】フレーム判定部12は、後述するフレームレート処理を行ってから（ステップS101）、現在の

フレームのピクチャ種別を判定する（ステップS10

2）。

【0031】現在のフレームがIピクチャである場合は、現在のフレームの画像データがキャプチャ部11のバッファ31から読み込まれ（ステップS111）、小ブロックの画素データが離散コサイン変換部15により空間周波数成分に変換され（ステップS112）、空間周波数成分が量子化器16により量子化され（ステップS113）、量子化された空間周波数成分が可変長符号化部17により可変長符号化され（ステップS114）、量子化された空間周波数成分が逆量子化器18により逆量子化され（ステップS115）、逆量子化された空間周波数成分より逆離散コサイン変換部19により各画素データが復元され（ステップS116）、復元された各画素データが参照フレームの各画素データとしてビデオメモリ21に書き込まれる（ステップS117）。ステップS112からステップS117までが、現在のフレームの全てのマクロブロックの全ての小ブロックについて実行される（ステップS118、ステップS119）。

【0032】現在のフレームがIピクチャである場合は、現在のフレームの画像データがキャプチャ部11のバッファ31から読み込まれ（ステップS131）、現在のフレームのマクロブロックとの差分が最小となる参照フレームのマクロブロックが動き検索部22により検索され（ステップS132）、現在のフレームのマクロブロックの画素データから参照フレームの検索されたマクロブロックの画素データが小ブロック単位で減算器14により引かれ（ステップS133）、減算器14が出力する差分が小ブロック毎に離散コサイン変換部15により空間周波数成分に変換され（ステップS134）、空間周波数成分が量子化器16により量子化され（ステップS135）、量子化された空間周波数成分が可変長符号化部17により可変長符号化され（ステップS136）、量子化された空間周波数成分が逆量子化器18により逆量子化され（ステップS137）、逆量子化された空間周波数成分より逆離散コサイン変換部19により差分が復元され（ステップS138）、復元された差分に参照フレームのデータが加算器20により加算され（ステップS139）、加算器20が出力する和が参照フレームの画素データとしてビデオメモリ21に書き込まれる（ステップS140）。ステップS133からステップS140までが、現在のマクロブロックの全ての小ブロックについて実行され（ステップS141）、ステップS132からステップS141までが、現在のフレームの全てのマクロブロックについて実行される（ステップS142）。

【0033】現在のフレームがBピクチャである場合は、同一フレーム符号生成部13が差分がないフレームであることを示す符号を生成する（ステップS15

1）。

【0034】ステップS119、ステップS142又はステップS151に続くステップS161では、生成した符号をマルチブレクサ25を通して出力する。ステップ161に続くステップS162では、GOP内の全フレームが終了したかどうかを判断して、終了していないければステップS101に戻り、終了していれば処理を終了する。

【0035】図3はフレーム判定部12が行うフレームレート処理（図2のステップS101）の詳細なフローチャートである。図3を参照すると、実質的なフレームレートが初期設定済みであるかどうかを判断して（ステップS201）、そうでない場合は実質的なフレームレートの初期設定を行う（ステップS202）。そうである場合は処理したフレーム数が $\alpha_1$ の倍数であるかどうかを判断して（ステップS203）、そうである場合は実質的なフレームレートの更新を行う（ステップS204）。ここで $\alpha_1$ の値は動画像符号化装置毎に異なる値であり、 $\alpha_1$ の単位は通常はGOPやシーケンスである。

【0036】図4はフレーム判定部12が行う現在のフレームがIピクチャであるときに、このフレームの画素データを読み込む処理（図2のステップS111）の詳細なフローチャートである。図4を参照すると、圧縮するIピクチャがキャプチャ済みであるかどうかを判断して（ステップS211）、そうでない場合はタイマ33により待ち時間を計測する（ステップS212）。そうである場合はキャプチャ部11のバッファ31より圧縮するIピクチャの原画像を読み込む（ステップS213）。

【0037】図5はフレーム判定部12が行う現在のフレームがPピクチャであるときに、このフレームの画素データを読み込む処理（図2のステップS131）の詳細なフローチャートである。図5を参照すると、圧縮するPピクチャがキャプチャ済みであるかどうかを判断して（ステップS221）、そうでない場合はタイマ33により待ち時間を計測する（ステップS222）。そうである場合はキャプチャ部11のバッファ31より圧縮するPピクチャの原画像を読み込む（ステップS223）。

【0038】図6は図3のフレームレートの初期設定（ステップS202）の詳細なフローチャートであり、図7は図3のフレームレートの更新（ステップS204）の詳細なフローチャートである。

【0039】フレームレートの初期設定の処理を図6を参照して説明する。

【0040】まず、以下の計算式により実質フレームレートの初期値を求める（ステップS301）。

【0041】実質フレームレートの初期値 =  $T [ (F1 \times (V2/V1) \times \alpha_2) ]$

9

但し、

F 1 : 基準画像の実質フレームレート

V 1 : 基準画像のサイズ

V 2 : 現在の画像サイズ

$\alpha$  2 : 補正值 ( $\alpha < 1$ )

ここで、T [] は、切り上げ、切り捨て又は四捨五入を表す。また、 $\alpha$  2 の値は圧縮処理にゆとりを持たせるための係数であり、動画像符号化装置毎に異なる。基準画像サイズと基準画像は、実質フレームレートが基準フレームレートとなるように定められたものである。

【0042】 例えば、

$$F_1 = 12.2$$

$$V_1 = 176 \times 144 = 25344$$

$$V_2 = 320 \times 240 = 76800$$

$$\alpha_2 = 0.8$$

の場合は、実質フレームレートの初期値は、T [] が切り捨てを表すとすれば、 $T [(12.2 \times 76800 / 25344) \times 0.8] = T [29.57] = 29$  となる。

【0043】 次に実質フレームレートとの最小公倍数が最も小さい出力フレームレートを選択する (ステップS 302)。この出力フレームレート [frame/sec] は 23.976, 24, 25, 29, 97, 30 の値から選択される。例えば実質フレームレートが 8 の場合は 24 となる。なお、処理を簡単にするために最も約数の多い 24 に固定しても良い。

【0044】 フレームレートの更新の処理を図7を参照して説明する。

【0045】 最後にフレームレートを更新してからのキャプチャ時の待ち時間の合計を求めて (ステップS 311)、最後にフレームレートを更新してからの挿入した差分がないフレーム (Bピクチャ) の数を求めて (ステップS 312)、現在の出力フレームレートを取り出して (ステップS 313)、以下の計算式により 1 フレーム当たりの圧縮処理時間 L を求める (ステップS 314)。

$$[0046] 1 \text{ フレーム当たりの圧縮処理時間 } L = ((F/R) - T) / (F - B) \times \alpha_3$$

但し、

F : 最後に実質フレームレートを更新してから圧縮したフレーム数

R : 現在の出力フレームレート

B : 最後に実質フレームレートを更新してから挿入した差分がないフレーム

$\mu$  (Bピクチャ) の数

T : 最後に実質フレームレートを更新してからのキャプチャ時の待ち時間の合計

$\alpha_3$  : 補正值 ( $\alpha_3 > 1$ )

この  $\alpha_3$  の値は圧縮処理にゆとりを持たせるための係数であり、動画像符号化装置毎に異なる。また、圧縮処理時間は、Iピクチャに対しても Pピクチャに対しても同

10  
一であると仮定しているが、Iピクチャの数に対して Pピクチャの数が多いので問題とはならない。

【0047】 例えば、

$$F = 49 \text{ [frame]}$$

$$R = 24 \text{ [frame/sec]}$$

$$B = 25 \text{ [frame]}$$

$$T = 160 \text{ [msec]}$$

$$\alpha_3 = 1.05$$

の場合は、1 フレーム当たりの圧縮処理時間 L は、

10  $(49/24 - 160/1000) / (49 - 25) \times 1.01 = 80.49 \text{ [msec/frame]}$  となる。

【0048】 次に 1 フレーム当たりの圧縮処理時間 L の値の逆数から実質フレームレートの値を計算して、結果が少数の場合は切り上げ／切り捨て／四捨五入を行い、整数にする (ステップS 315)。例えば、

$$L = 80.49 \text{ [msec/frame]}$$

の場合は、更新した実質フレームレートは、

$$1000 / 80.49 \text{ [frame/sec]}$$

の、小数点を切り捨てて 12 [frame/sec] となる。

20 【0049】 次に、実質フレームレートとの最小公倍数が最も小さい出力フレームレートを選択する (ステップS 316)。この出力フレームレート [frame/sec] は 23.976, 24, 25, 29, 97, 30 の値から選択される。例えば実質フレームレートが 8 の場合は 24 となる。なお、処理を簡単にするために最も約数の多い 24 に固定しても良い。

【0050】 また、図8はピクチャ種別を判定する処理 (図2のステップS 102) のフローチャートである。図8では最後に実質フレームレート及び出力フレームレートを更新してから圧縮したフレーム数を求めて (ステップS 401)、現在の実質フレームレートを取り出して (ステップS 402)、現在の出力フレームレートを取り出す (ステップS 403)。次にメモリ32に格納している Bカウンタの値 (初期値は 0) が 0 以上であるかどうかを判断して (ステップS 404)、そうでない場合は以下の計算式から次回に I フレーム又は P フレームとの間に挿入する B ピクチャのフレーム数を計算して Bカウンタに格納する (S 75)。

【0051】 Bピクチャのフレーム数 =  $T \times (K \times R_1 / R_2) - N - 1$

40 但し、

N : 最後にフレームレートを更新してから圧縮したフレーム数

K : Bピクチャのフレーム数の計算の計算回数 (フレームレートを更新する度に 1 から始まる)

R 1 : 現在の出力フレームレート

R 2 : 現在の実質フレームレート

ここで、T [] は、切り上げ、切り捨て又は四捨五入を表す。

50 【0052】 例えば、

11

 $N = 4$  $K = 3$  $R_1 = 24$  $R_2 = 12$ 

の場合は、次回にIフレーム又はPフレームとの間に挿入するBピクチャのフレーム数は、

$$(3 \times 24 / 12) - 4 - 1 = 1$$

となる。

【0053】次に、最後にフレームレートを更新してから圧縮したフレーム数が出力フレームレートの倍数であるかどうかを判断する（ステップS406）。そうである場合は圧縮するピクチャ種別をIピクチャにする（ステップS407）。また、倍数でない場合は圧縮するピクチャ種別をPピクチャにする（ステップS408）。こうすることにより1秒に1回だけフレームはIピクチャとなる。従ってステップs406の判定基準を変えることにより、Iピクチャのレートを変えることが出来る。

【0054】また、Bカウンタの値が0以上である場合は圧縮するピクチャ種別をBピクチャにして（ステップS409）、Bカウンタを1減算する（ステップS410）。

【0055】次に具体例を用いて説明する。

【0056】図9はBピクチャの処理の負荷を削除することでリアルタイムに圧縮できることを説明する図である。図9では縦に積まれた棒はIピクチャやPピクチャの圧縮処理時間の合計を示していて、各ピクチャの圧縮処理時間の合計が実質フレームレートに応じた時間／フレームの右上がりの直線よりも下にあればリアルタイムで処理できることになる。フレーム番号毎に圧縮処理を説明する。

【0057】1フレーム目

Iピクチャの圧縮なのでIピクチャの圧縮を始める。

【0058】2フレーム目

本来はBピクチャの圧縮を行うが、まだIピクチャの圧縮が終了していないのでIピクチャの圧縮を継続する。

【0059】3フレーム目

本来はBピクチャの圧縮を行うが、まだIピクチャの圧縮が終了していないのでIピクチャの圧縮を継続する。なお、Iピクチャの圧縮が終了後に2つのBピクチャを実際には圧縮しないで差分がないフレームであることを示す符号を作成する。この時の処理時間はほとんど無視できる。

【0060】4フレーム目

Pピクチャを圧縮するが、まだそのフレームをキャプチャしていないので、キャプチャするのを待ってから圧縮を始める。この待ち時間がキャプチャ待ち時間である。

【0061】5フレーム目

本来はBピクチャの圧縮を行うが、まだPピクチャの圧縮が終了していないのでPピクチャの圧縮を継続する。

12

### 【0062】6フレーム目

本来はBピクチャの圧縮を行うが、まだPピクチャの圧縮が終了していないのでPピクチャの圧縮を継続する。なお、Pピクチャの圧縮が終了後に2つのBピクチャを実際には圧縮しないで差分がないフレームであることを示す符号を作成する。この時の処理時間はほとんど無視できる。

### 【0063】7フレーム目

Pピクチャを圧縮するが、まだそのフレームをキャプチャしていないので、キャプチャするのを待ってから圧縮を始める。以降、同様にして圧縮を繰り返す。

【0064】このように圧縮性能が低くて1フレームの圧縮処理時間が長くかかる場合、Bピクチャの圧縮処理時間を削除できるのでリアルタイムに圧縮できる。

【0065】なお、図8にしめす動作により、2フレーム目、3フレーム目、5フレーム目、6フレームはBピクチャとして扱われる。

【0066】また、図10はGOP構成を説明する図である。図10の例では出力フレームレートが24[frame/sec]で実質フレームレートが9[frame/sec]となっている。具体的には以下に示す計算式で与えられる値Gを使用して、IピクチャとPピクチャが均等に散らばるように配置する。

$$[0067] G = T [N \times R_1 / R_2]$$

但し、

N：今までに圧縮したフレーム数

R1：出力フレームレート

R2：実質フレームレート

ここで、T[]は、切り上げ、切り捨て又は四捨五入を表す。

【0068】値Gは、Iピクチャ又はPピクチャのフレーム番号となる。値Gが実質フレームレートの倍数である場合はIピクチャのフレーム番号として、そうでない場合はPピクチャのフレーム番号とすることにより1秒に1回だけフレームはIピクチャとなる。

【0069】また、MPEG1の出力フレームレートには23.976[frame/sec]、24[frame/sec]、25[frame/sec]、29.97[frame/sec]、30.0[frame/sec]の数種類あるが、圧縮効率を上げるためにBピクチャの数がなるべく少なくなるようにするには実質フレームレートとの最小公倍数が最も小さい出力フレームレートを選択するのが良い。つまり、実質フレームレートが2[frame/sec]、3[frame/sec]、4[frame/sec]、6[frame/sec]、8[frame/sec]、12[frame/sec]の場合は出力フレームレートは24[frame/sec]を選択して、実質フレームレートが5[frame/sec]の場合は出力フレームレートは25[frame/sec]を選択するのが良い。ただし、処理を簡単にするために出力フレームレートを最も約数の多い24[frame/sec]に固定しても良い。

【0070】図10の例ではN=0の場合は $0 \times 24 / 9 = 0$ の

で0番目のフレームはIピクチャとなり、N=1の場合は $1 \times 24 / 9 = 2.666$ なので四捨五入して3番目のフレームはPピクチャとなり、その間の1~2番目のフレームはBピクチャとなる。次にN=2の場合は $2 \times 24 / 9 = 5.33$ なので四捨五入して5番目のフレームはPピクチャとなり、その後の4番目のフレームはBピクチャとなる。これ以降も同様な計算を繰り返してピクチャ種別を決める。

【0071】また、図11に差分データが全くない時のBピクチャの符号列例を示す。図11のアルファベットは符号の種類を表し、アルファベットの右側のビット数は符号のビット数を表し、ビット数の右側の数値は符号の値を示す。末尾に(H)が付いている値は16進数の値であり、末尾に(D)が付いている値は10進数の値であり、末尾に何も付いていない値は2進数の値である。Bピクチャの符号列は、フレームの先頭を示すPSC(Picture Start Code)の符号から始まる。符号TRの後のPCT(Picture Coding Type)の値は011(B)であり、この値は、現在のフレームがBピクチャであることが示している。圧縮データはマクロブロック単位で記述する。MBAI(Macro Block Address Increment)は前のマクロブロックからの増加分を示し、先頭のMBAIの値は1である。MBAIに続くMBTYPE(Macro Block Type)はマクロブロックの形式を記述するが、そのうちのMBPの値が0ということはそのマクロブロックには符号化するデータがないことを示している。MBTYPEの値である0010は可変長符号の値であり、図11に示すMBTYPEに含まれる5つの要素の値が符号化されたときの値である。MBAI~MVFに続くMBESC(Macro Block Escape)は33個分のマクロブロックに符号化データがないことが示している。これが2回記述されているので66個分のマクロブロックに符号化データがないことが分かる。続くMBAIの値は13であり、この値は、さらに12個のマクロブロックには符号化データがないことをしめす。MBAIに続くMBYTE中のMBPの値の0は、スキップした12個のマクロブロックに続く13個目のマクロブロックにも符号化データがないことを示す。従って符号化されないマクロブロック数は $1 + 2 \times 33 + 13 = 80$ であるが、この例で示したフレームは80個のマクロブロックより構成されているので、図11の符号列は、フレーム中の全てのマクロブロックが符号化されていないことを示している。このBフレームの符号列のビット数は、2つのMBESCとこれらを挟む符号のビット数を合わせて、 $2 \times 11 + 135 = 157$ ビットとなる。

【0072】また、MBESCの数を増減することにより任意の画像サイズに応じて差分がないフレームであることを示す符号を生成することができる。MBESCの数は以下の計算式から求められる。

【0073】

$$\text{MBESCの数} = (\text{MB} - \text{MB1} - \text{MB2}) / \text{MB3}$$

但し、

MB : 1画面分のマクロブロックの個数

MB1 : 先頭のMBAI~MVFで示されるマクロブロックの数(1個)

MB2 : 最後のMBAI~MVFで示されるマクロブロックの最大数(33個)

MB3 : 1個のMBESCで示されるマクロブロックの数(33個)

また、最後のMBAIはスキップするマクロブロック数により符号が異なるので最大ビット数は11ビットとなり、MBESCを除いた最大ビット数は142ビットとなる。

【0074】このように差分がないフレームであることを示す符号を作成する処理は非常に単純な処理なので、実際に圧縮する処理に比べて無視できるほど負荷が軽い。

【0075】

【発明の効果】以上説明したように、本発明によれば、動画像符号化装置の性能に合わせて高能率に圧縮できることにある。何故ならば、両方向フレーム間圧縮を行うフレームが差分がないフレームであることを示す符号を生成することにより実質的なフレームレートを動画像符号化装置がリアルタイムで圧縮できるフレームレートに合わせることができるからである。

【0076】また、本発明によれば、動画像符号化装置の動作中に圧縮性能が変わってもそれに合わせて実質的なフレームレートを変えられることにある。何故ならば、両方向フレーム間圧縮を行うフレームが差分がないフレームであることを示す符号の割合をキャプチャ時の待ち時間とBピクチャを挿入したフレーム数等から再計算して動的に変えることができるからである。

【図面の簡単な説明】

【図1】本発明の実施形態による動画像符号化装置の構成を示すブロック図である。

【図2】本発明の実施形態による動画像符号化装置の動作を示すフローチャートである。

【図3】図2のフレームレート処理の詳細を示すフローチャートである。

【図4】図2のIピクチャの読み込みの詳細を示すフローチャートである。

【図5】図2のPピクチャの読み込みの詳細を示すフローチャートである。

【図6】図3のフレームレートの初期設定の詳細を示すフローチャートである。

【図7】図3のフレームレートの更新の詳細を示すフローチャートである。

【図8】図2の現在のフレームのピクチャ判別の詳細を示すフローチャートである。

【図9】本発明の実施形態によるBピクチャの処理の負荷を削除することでリアルタイムに圧縮できることを説

15

明する図である。

【図10】本発明の実施形態によるG O P構成を説明する図である。

【図11】差分データが全くないBピクチャの符号列の例を示す図である。

【図12】M P E G方式の符号の階層構造を示す図である。

【図13】M P E G方式に準拠した符号フォーマットの階層図を示した図である。

【符号の説明】

1 1 キャプチャ部

1 2 フレーム判定部

1 3 同一フレーム符号生成部

16

\* 1 4 減算器

1 5 離散コサイン変換部

1 6 量子化器

1 7 可変長符号化部

1 8 逆量子化器

1 9 逆離散コサイン変換部

2 0 加算器

2 1 ビデオメモリ

2 2 動き検索部

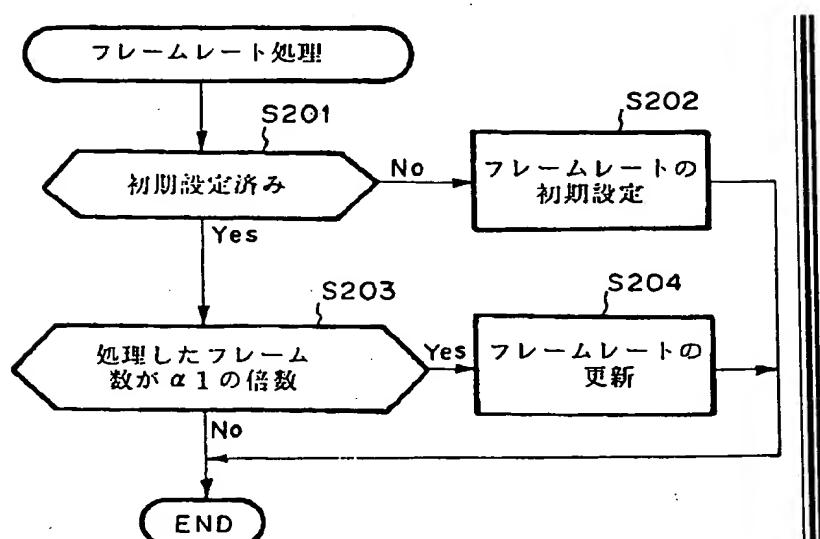
10 2 3 動き補償部

2 4 スイッチ

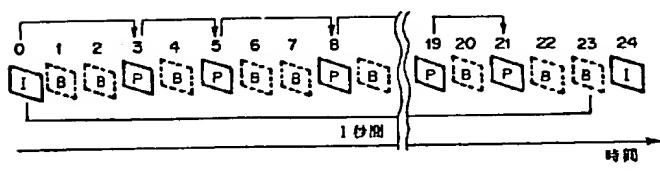
2 5 マルチブレクサ

\* 4 1 符号化部

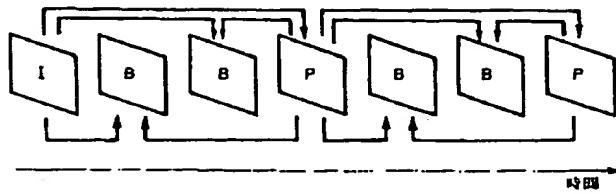
【図3】



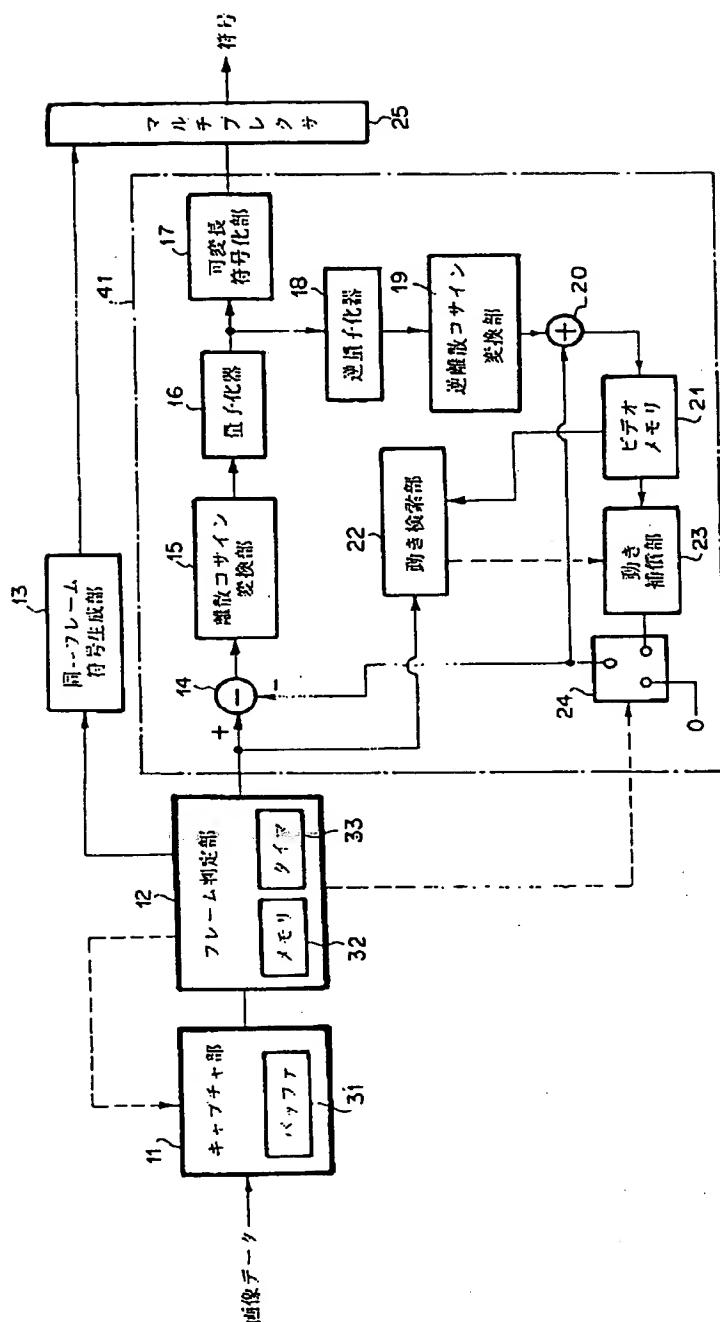
【図10】



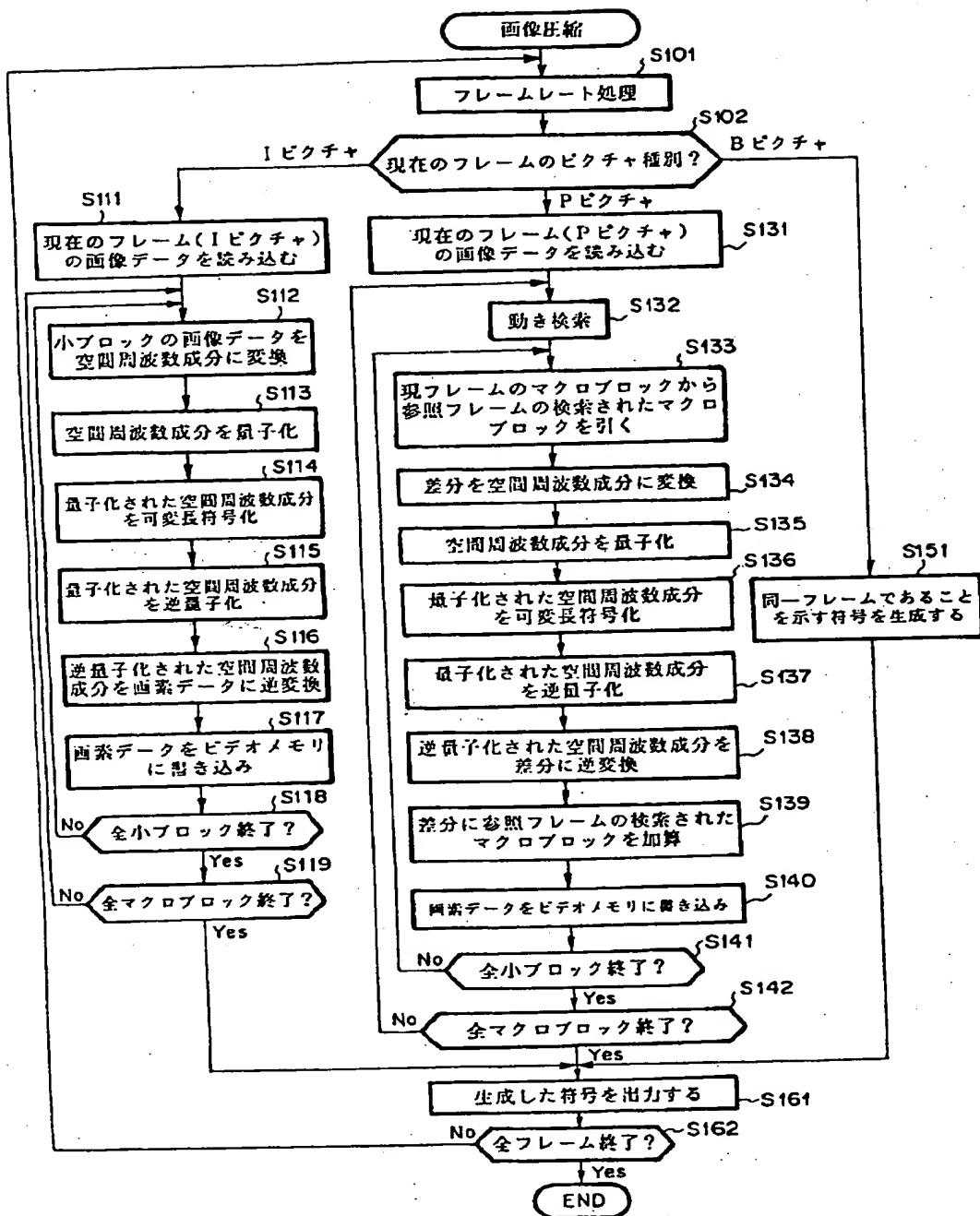
【図13】



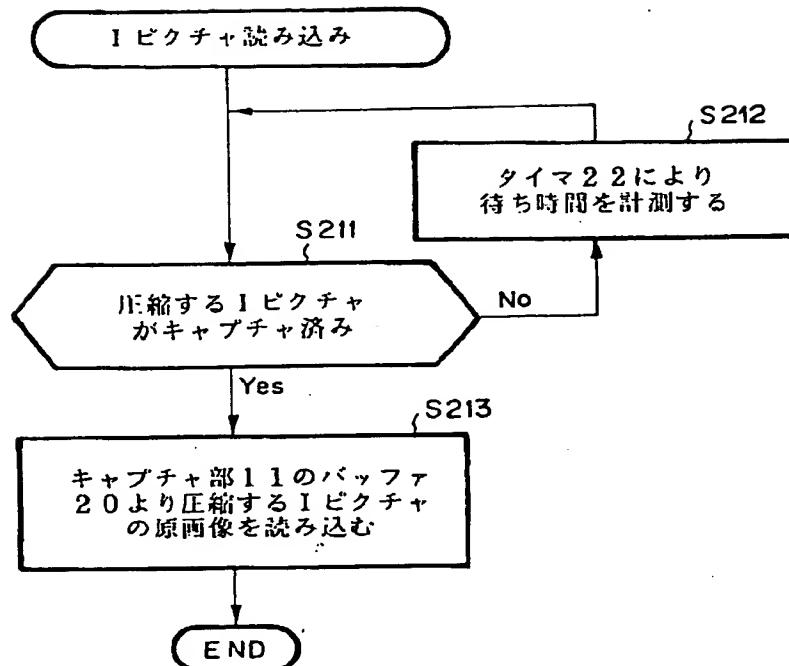
【図1】



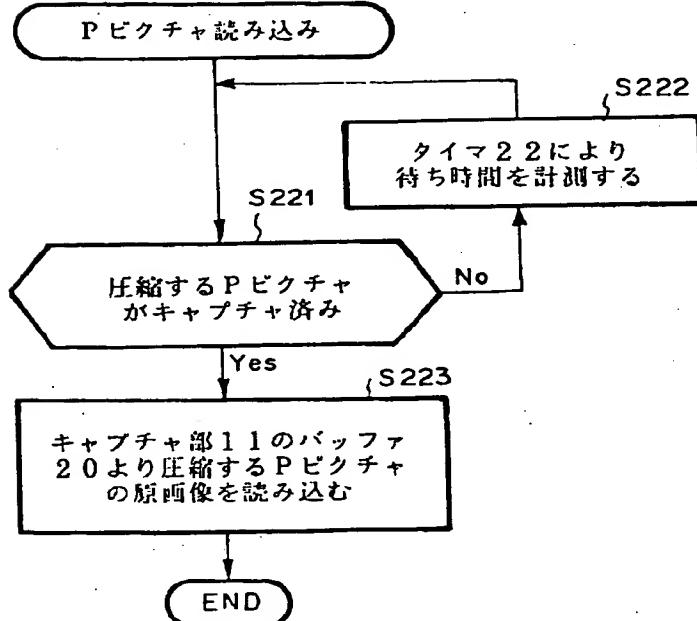
【図2】



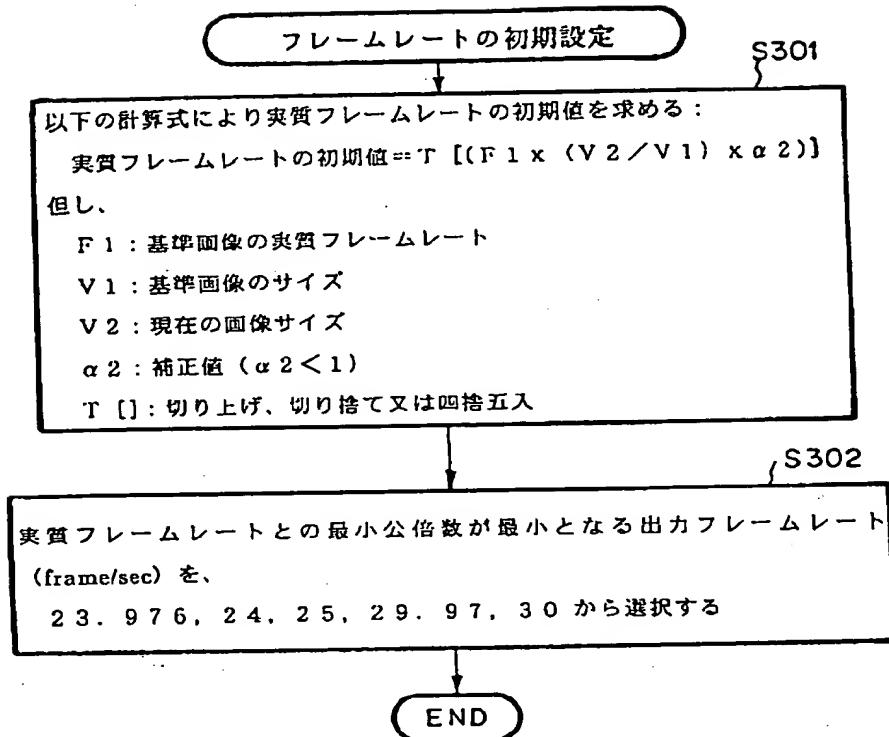
【図4】



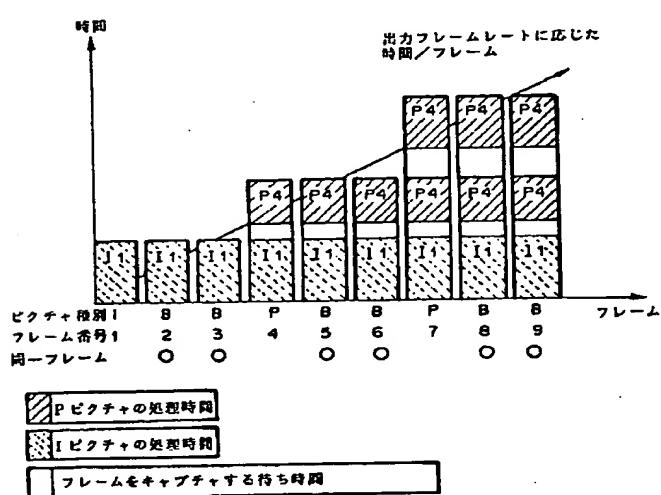
【図5】



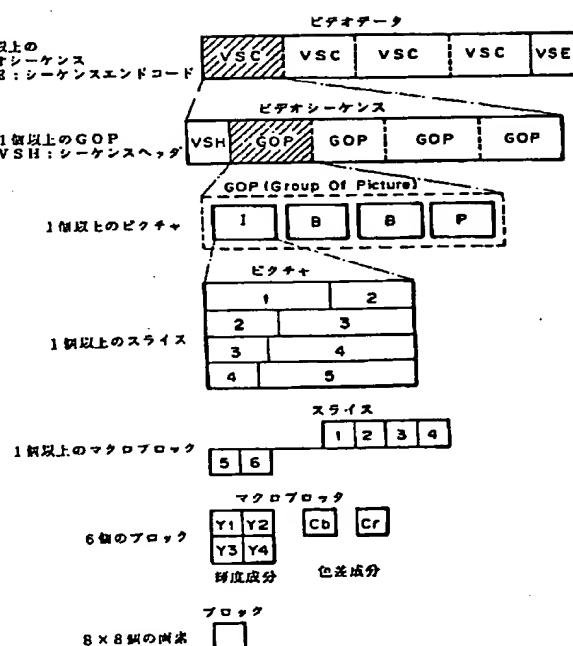
【図6】



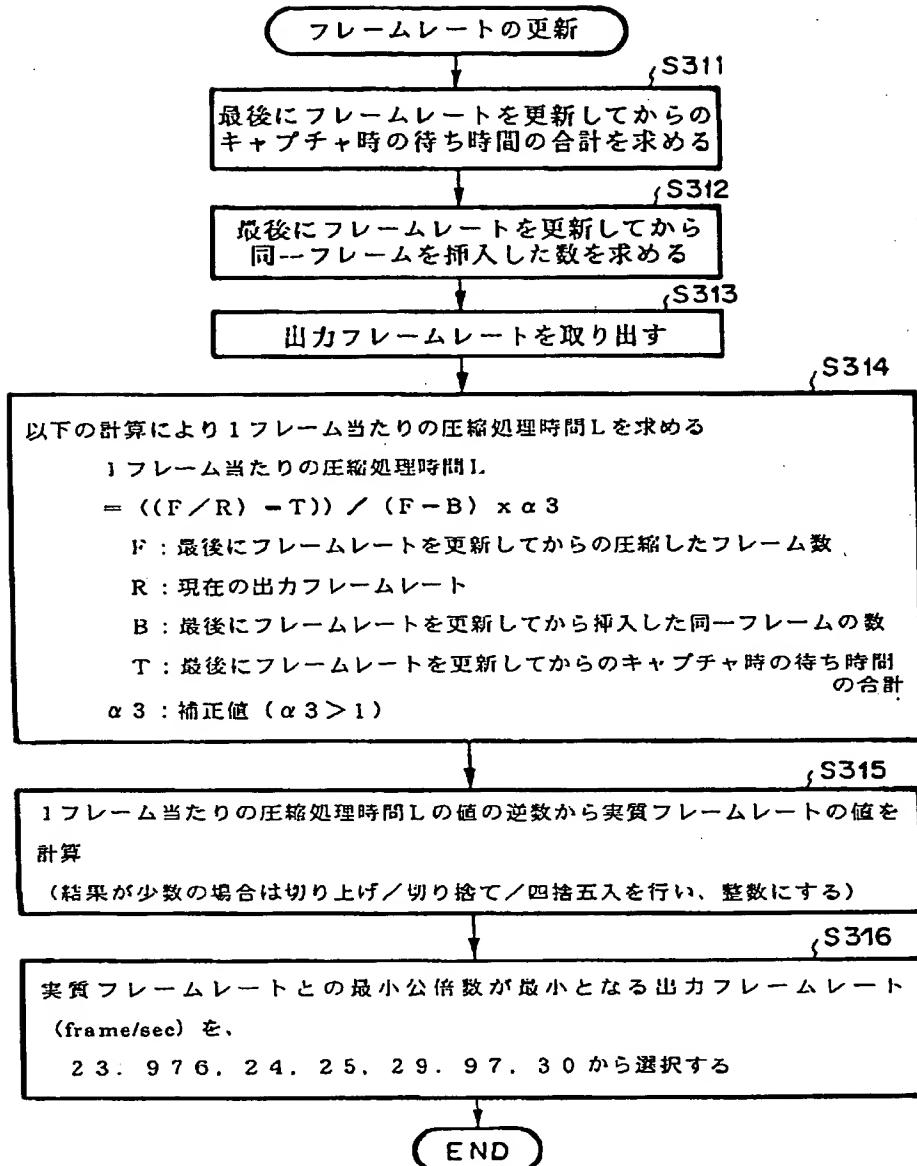
【図9】



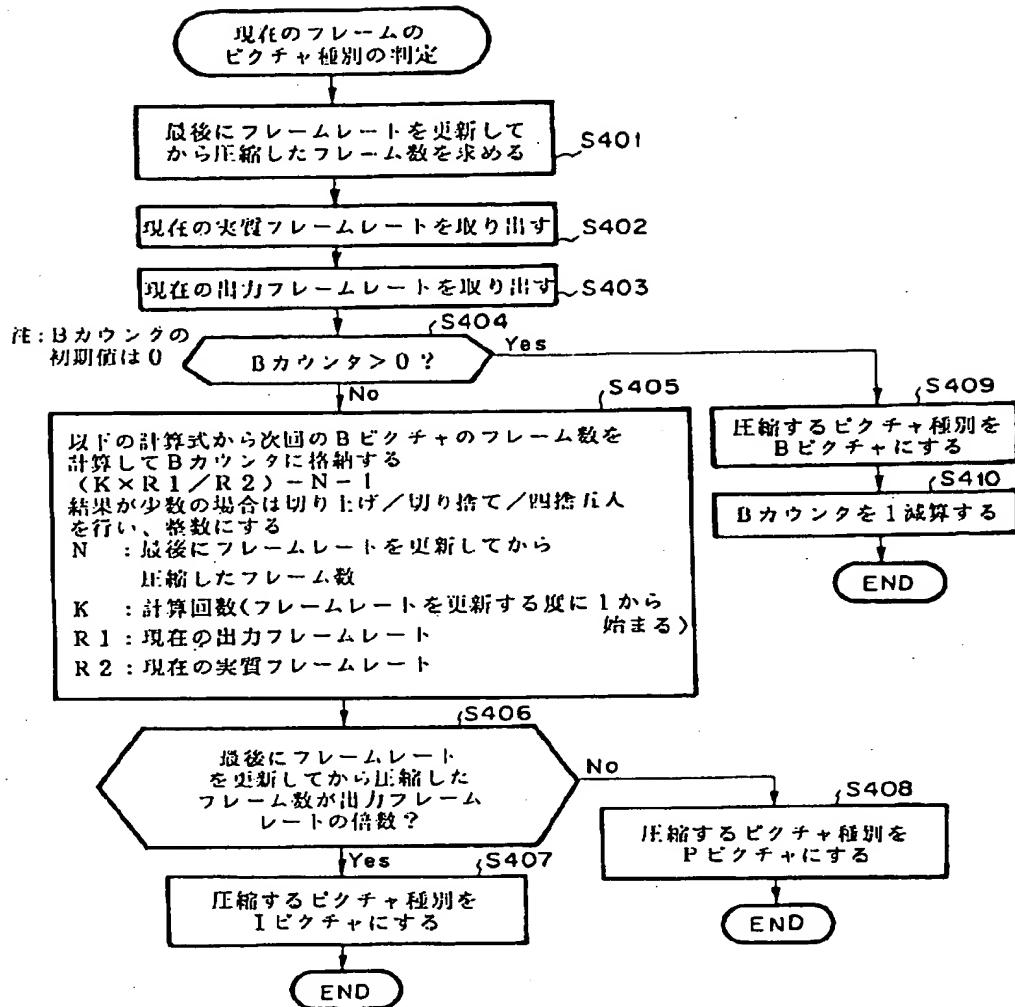
【図12】



【図7】



[図 8]



【図11】

PSC	32bit	00000100 (H)	EPM TC 2600 FINAL SEARCH DATE 1-6-04 DELIVER TO GOV'T DATE
TR	10bit	1	
PCT	3bit	011	
VD	16bit	48000 (D)	
FPPV	1bit	0	
FFC	3bit	001	
FPPV	1bit	0	
BFC	3bit	011	
EBP	1bit	0	
SSC	32bit	000001AF (H)	画像サイズに よらず固定
OS	5bit	18 (D)	
EBS	1bit	0	
MBAI	1bit	1	
MBTYPE	4bit	0010	MBTYPE
QS	5bit	0	MBQUANT 1bit 0
MHF	1bit	0	MDFV 1bit 1
MVF	1bit	0	MDBV 1bit 0
MB ESC	11bit	0000 0001 000	MBP 1bit 0
MB ESC	11bit	0000 0001 000	MBINTRA 1bit 0
MBAI	4bit	13 (D)	
MBTYPE	4bit	0010	画像サイズによって MBAI の符号が変わる
QS	5bit	0	
MHF	1bit	0	
MVF	1bit	0	
PSC :Picture Start Code			フレームの先頭を示す
TR :Temporal Reference			フレームの番号 (GOP 每にリセット)
PCT :Picture Coding Type			フレーム種別
VD :VBV Delay			ランダムアクセス時のバッファの初期状態
FPPV :Full Pel Forward Vector			順方向動きベクトルの精度 (全画素/半画素)
FFC :Forward f Code			順方向動きベクトルの記述範囲
FPPV :Full Pel Backward Vector			逆方向動きベクトルの精度 (全画素/半画素)
BFC :Backward f Code			逆方向動きベクトルの記述範囲
EBP :Extra Bit Picture			拡張用の情報の有る/無しを示すビット
SSC :Slice Start Code			スライスの先頭を示す
OS :Quantizer Scale			量子化で使用される値
EBS :Extra Bit Slice			拡張用の情報の有る/無しを示すビット
MBAI :Macroblock Address Increment			その前にスキップする MB の数を示す
MBTYPE :Macroblock Type			マクロブロック (MB) の符号化種別を示す
MHF :Motion Horizontal Forward Code			その MB の順方向動きベクトルの水平成分と 前の MB のベクトルとの差分を示す
MVF :Motion Vertical Forward Code			MHF と同様に順方向動きベクトルの垂直成分を示す
MB ESC :Macroblock Escape			MB 3 3 個スキップに相当する